Learning maximum excluding ellipsoids from imbalanced data with theoretical guarantees

Guillaume Metzler, Xavier Badiche, Brahim Belkasmi, Elisa Fromont, Amaury Habrard, Marc Sebban

Learning Maximum Excluding Ellipsoids from Imbalanced Data with Theoretical Guarantees

Guillaume Metzler[1,3], Xavier Badiche [1,3], Brahim Belkasmi[1], Elisa Fromont[2], Amaury Habrard[1], and Marc Sebban[1]

[1] Laboratoire Hubert Curien UMR 5516, Univ Lyon, UJM-Saint-Etienne, F-42023, Saint-Etienne,
[2]IRISA/Inria, Univ. Rennes 1, 35042 Rennes cedex, France
[3] Blitz Business Service inc., France.

Abstract

In this paper, we address the problem of learning from imbalanced data. We consider the scenario where the number of negative examples is much larger than the number of positive ones. We propose a theoretically-founded method which learns a set of local ellipsoids centered at the minority class examples while excluding the negative examples of the majority class. We address this task from a Mahalanobis-like metric learning point of view and we derive generalization guarantees on the learned metric using the uniform stability framework. Our experimental evaluation on classic benchmarks and on a proprietary dataset in bank fraud detection shows the effectiveness of our approach, particularly when the imbalancy is huge.

1 Introduction

The study of imbalanced data is an active and important topic due to its huge economical impact, for example in anomaly or fraud detection in applications related to banks, medicine, intrusion detection, video-surveillance, fault detection or industrial processes (Agrawal and Agrawal, 2015; Khalilia et al., 2011; Chandola et al., 2009; Goldstein and Uchida, 2016). In such applications, datasets are usually composed of a large number of negative examples (e.g. genuine transactions, normal MRI, normal human behavior, etc.) and only a few positive data (e.g. frauds, faults, hacking, etc.). From a theoretical point of view, imbalanced scenarios raise two main challenging problems for the machine learning and data mining communities. First, the large majority of classic supervised learning methods optimize the accuracy by minimizing error-based loss functions, like the hinge loss in Support Vector Machines, the exponential loss in Boosting, or the logistic loss in Logistic Regression. However, if one class is rare, those methods will struggle to capture any useful information about this class and will obtain a high accuracy by simply predicting all (possibly new) examples as being of the majority class(es). Second, deriving theoretical results in an imbalanced setting is difficult and many existing approaches come without any specific guarantee.

In this paper, we aim at addressing both issues by designing a new algorithm - driven by the minority class - for which generalization guarantees are derived. Unlike the state of the art, our method does not resort to sampling preprocesses. Indeed, it is worth noticing that classic methods typically consist of over/under-sampling the data (Aggarwal, 2013) or creating synthetic examples in the neighborhood of the minority class (Chawla et al., 2002) in order to get more balanced sets.

However, as we will see in this paper, when the datasets become highly imbalanced (i.e. no more than 2-3% positives), those methods usually fail to model the minority class because they suffer from an inability to generate enough diversity which is key for a sampling method to work well. In such extreme situations, two main families of solutions are available. The first one consists of designing **alternative loss functions** able to directly capture the imbalance in the data and usually based on the F-Measure, the area under the ROC curve, or the Average Precision (Frery et al., 2017). The main pitfalls related to this line of research concern the difficulty to deal with non smoothed and non convex measures. The second category of methods aims at changing the original problem into an **unsupervised** anomaly detection task. This is for example done by Support Vector Data Description (SVDD) (Azami et al., 2014; Pauwels and Ambekar, 2011; Tax and Duin, 2004) methods such as one-class SVMs (OCSVM) (Heller et al., 2003) which work on unlabeled data. SVDD learns the smallest enclosing ball which includes most of the training data and excludes all examples lying in the tail of the data distribution, which are considered as anomalies. More formally, the goal is to solve the following constrained optimization problem given a sample of n instances:

$$\begin{array}{ll} \underset{R,\boldsymbol{c},\boldsymbol{\xi}}{min} & R^2 + \frac{\mu}{n}\sum_{i=1}^{n}\xi_i, \\ s.t. & \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq R^2 + \xi_i, \quad \forall i = 1, \ldots, n, \\ & \xi_i \geq 0, \end{array} \tag{1}$$

where R and $\mathbf{c}$ are respectively the radius and the center of the ball and ξ_i is the slack variable associated to the i^{th} instance $\boldsymbol{x}_i$. μ is tuned in order to control the proportion of the data outside the sphere (considered as anomalies). Note that in Pauwels and Ambekar (2011), the authors have shown that using the radius instead of the square of the radius in this formulation is often preferable. Several refinements of the SVDD method can be found in the literature to take more than one class into account (e.g. see Liu and Zheng (2006) and Boujnouni et al. (2012) for binary problems) or to use multiple local models and apply non linear transformation to the data (Le et al., 2013; Boujnouni et al., 2012). Even if the kernel-based methods are effective, the computation of the kernel is often expensive (according to the number of examples in the dataset) and does not scale well on most real datasets. An interesting approach, which does not suffer from this drawback, is presented in Wang et al. (2010). The authors include a linear transformation of the data, in the form of a Positive Semi Definite (PSD) matrix $\mathbf{M}$ in the SVDD optimization problem. To avoid high computational costs, they set $\mathbf{M}$ to be the inverse of the covariance matrix, which allows the induction of ellipsoids rather than spheres. Such objects are able to cover a larger volume in the input space compared to the spheres.

In this paper, we present a supervised machine learning method called ME^2 - for Maximum Excluding Ellipsoids - that can tackle imbalanced data without requiring sampling strategies. We exploit the idea of learning multiple local models to capture non linearity at a cheap cost as in Le et al. (2013) and combine it with a **metric learning** formulation. However, we learn local models centered at each minority class example which exclude the examples of the majority class(es). This, contrary to the previously cited methods, allows us to consider settings where the minority class examples do not necessarily behave as anomalies but may be hidden within one of the modes of the data (e.g. in fraud detection). Unlike Wang et al. (2010), we optimize both the shape and the orientation of the ellipsoid by learning a Mahalanobis distance based on a PSD matrix $\mathbf{M}$. This geometrical flexibility allows us to capture more accurate and potentially larger areas around the minority class examples. A nice property of our approach is that $\mathbf{M}$ can be obtained in closed-form solution ensuring directly (and for free) the positive definiteness of $\mathbf{M}$ (Shi et al., 2014; Perrot and Habrard, 2015). Therefore, we prevent the algorithm from having to check the positiveness of the eigenvalues of $\mathbf{M}$, which has a cubic complexity in the size of $\mathbf{M}$, as required by many metric learning algorithms (Bellet et al.,

2013, 2015). Beyond its algorithmic contribution, materialized by ME^2, this paper also aims at providing guarantees on the learned models based on the uniform stability framework (Bousquet and Elisseeff, 2002). We prove that our algorithm is stable, i.e. robust to changes in the training set.

To sum up, our contribution is four-fold:

1. We introduce a **simple strategy** consisting of learning local models centered at each positive example (which can be done in parallel) allowing us to capture non linearity in the feature space. This way, our algorithm is particularly relevant in the context of imbalanced settings where (i) the number of local models to learn is very small but (ii) the minority examples (i.e. the centers of the ellipsoids) play a key role.

2. We show that ME^2 is **algorithmically efficient** by demonstrating that the learned matrices satisfy the PSD constraint for free.

3. We prove that ME^2 is **theoretically founded** by deriving generalization guarantees on the learned models.

4. We experimentally show that ME^2 is **effective** compared to the state of the art methods, particularly on highly imbalanced settings.

As far as we know, ME^2 is the first method able to gather all those interesting features in the field of learning from imbalanced data.

The rest of this paper is organized as follows: In Section 2, we present the convex formulation of our algorithm ME^2 with both primal and dual formulations. Section 3 is devoted to the theoretical study of our algorithm. We show that ME^2 is stable which provides some insight into the variance of the algorithm with respect to changes in the training set. In Section 4, we compare our algorithm to some state-of-the-art methods. We conclude in Section 5.

2 ME^2: a metric learning-based algorithm for optimizing excluding ellipsoids

2.1 Problem formulation

Let $S = \{\boldsymbol{x}_i\}_{i=1}^n$ be a sample of n *negative* instances (the majority class) and $P = \{\boldsymbol{c}_j\}_{j=1}^p$ a set of p *positive* examples (the minority class), with $n >> p$ and where $\boldsymbol{x}_i, \boldsymbol{c}_j$ are feature vectors of $\mathbb{R}^d$. We aim at maximizing ellipsoids centered at each positive $\boldsymbol{c} \in P$ excluding (most of) the negative data $\boldsymbol{x}_i,\ i = 1, ..., n$. Learning such ellipsoids boils down to optimizing a Mahalanobis distance, that is finding a positive semi-definite (PSD) $d \times d$ matrix $\mathbf{M}$ projecting the data linearly in a new space and allowing to obtain balls centered at each positive example of maximum radius R. Let B be an upper bound of the possible expected radius. Our algorithm, called ME^2 for Maximum Excluding Ellipsoids, can be expressed in the following form:

$$\begin{array}{ll} \underset{R,\boldsymbol{M},\boldsymbol{\xi}}{min} & \frac{1}{n}\sum_{i=1}^n \xi_i + \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2, \\ s.t. & \|\boldsymbol{x}_i - \boldsymbol{c}\|_{\mathbf{M}}^2 \geq R - \xi_i, \quad \forall i = 1, \ldots, n, \\ & \xi_i \geq 0, \\ & B \geq R \geq 0, \end{array} \tag{2}$$

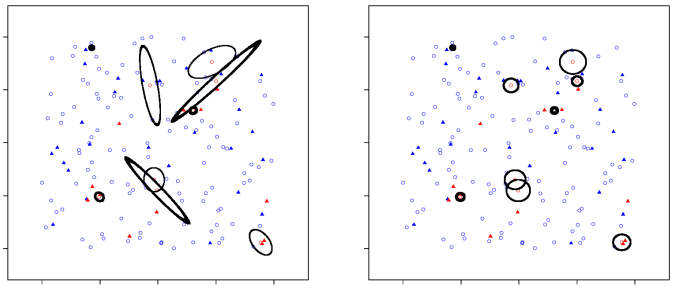

Figure 1: Illustration of the interest of learning ellipsoids (on the left) rather than simple spheres (on the right). Optimizing the size and the orientation of the ellipsoids allows us to better capture local peculiarities.

where $\|\boldsymbol{x}_i - \boldsymbol{c}\|^2_{\mathbf{M}} = (\boldsymbol{x}_i - \boldsymbol{c})^T\mathbf{M}(\boldsymbol{x}_i - \boldsymbol{c})$ is the learned Mahalanobis distance, $\boldsymbol{\xi}$ is the vector of slack variables, $\mu(B-R)^2 + \lambda\|\mathbf{M} - \mathbf{I}\|^2_F$ is a regularization term with $\mu, \lambda > 0$ the corresponding regularization parameters, $\mathbf{I}$ the Identity matrix and $\|\cdot\|_F$ is the Frobenius nom. Note that the upper bound B of the radius is used in $\mu(B-R)^2$ to have a convex formulation allowing us to get a unique solution. We choose two different parameters for each part of the regularization term to control the surface area of the sphere in the transformed space and the complexity of the matrix $\mathbf{M}$. The parameter λ gives the possibility to control the magnitude of the entries of $\mathbf{M}$, and therefore the shape/orientation of the ellipsoid. In practice, the bigger λ is, the closer $\|\boldsymbol{x}_i - \boldsymbol{c}\|^2_{\mathbf{M}}$ to the Euclidean distance (i.e. the ellipsoid looks like a ball). On the other hand, the parameter μ controls the size of the learned ellipsoids.

An illustration of our algorithm is given in Figure 1. On the right, we constrain ME^2 to learn spheres (i.e. λ is set to a large value such that $\mathbf{M}$ tends to be the identity matrix). On the left, we allow ME^2 to optimize both the orientation and the size of the ellipsoid. We can see that ME^2 can capture local peculiarities of the feature space. It is worth noticing that λ and μ are key parameters to deal with anomaly/fraud detection in imbalanced settings. Indeed, we will show that they can be used to improve the F-Measure by controlling the precision and the recall.

Note that we can establish a relationship between ME^2 and a decision tree algorithm (Quinlan, 1993). Indeed, in both cases, decision rules take the form of local geometric shapes (an ellipsoid for ME^2 and a rectangle for a decision tree). In Figure 2, we report on the same toy example as in Figure 1 the leaves learned by a decision tree algorithm containing each positive example as well as the ellipsoids optimized by ME^2. We can notice that while decision trees build axis-parallel hyperplanes to generate the leaves, ME^2 has a better expressiveness allowing to control the shape, the orientation and the size of the ellipsoids. We think that this is an interesting feature that can be favorably exploited to capture local specificities of the feature space and better estimate the density function of the positives.

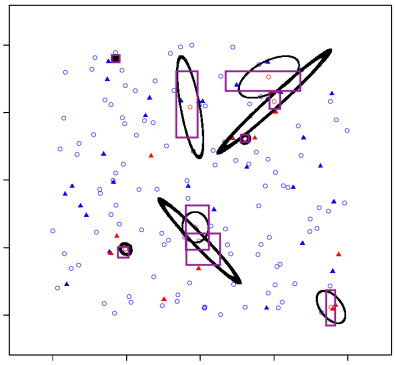

Figure 2: Boundaries of the decision rules with ME^2 and a decision tree algorithm. The expressiveness of ME^2 is better to capture local specificities of the density function of the positives.

2.2 Dual version and closed-form solution

Note that Problem 2 can also be expressed in its dual form which leads to a closed form solution. The Lagrangian is given by:

$$\mathcal{L}(\boldsymbol{\alpha}, \beta, \delta, \boldsymbol{\gamma}, R, \boldsymbol{\xi}, \mathbf{M}) = \frac{1}{n}\sum_{i=1}^{n}\xi_i + \mu(B-R)^2 - \sum_{i=1}^{n}\gamma_i\xi_i - \sum_{i=1}^{n}\alpha_i\left(\|\boldsymbol{x}_i - \boldsymbol{c}\|^2_{\mathbf{M}} - R + \xi_i\right) + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2 - \beta R + \delta(R-B), \quad (3)$$

where $\boldsymbol{\alpha} = (\alpha_i)_{i=1,\ldots,n}$, $\boldsymbol{\gamma} = (\gamma_i)_{i=1,\ldots,n}$, β and δ are the dual variables. By setting to zero all the derivatives of (3) with respect to the primal variables we get:

$$\begin{aligned} R &= \frac{\beta - \delta + 2\mu B - \sum_{i=1}^{n}\alpha_i}{2\mu}, \\ \mathbf{M} &= I + \frac{1}{2\lambda}\sum_{k=1}^{n}\alpha_k(\boldsymbol{x}_k - \boldsymbol{c})(\boldsymbol{x}_k - \boldsymbol{c})^T, \\ \textit{and} \qquad & 0 \leq \alpha_i \leq \frac{1}{n} \quad \forall i. \end{aligned}$$

The second equality shows that $\mathbf{M}$ is, by construction, *positive definite* (PD). Fulfilling the PD constraint for free is very important because it prevents the algorithm from performing a singular value decomposition (in $\mathcal{O}(d^3)$) at each step of the gradient descent.

The dual formulation of Problem (2) is then obtained by injecting the expression of both R and $\mathbf{M}$ in the Lagrangian (3). The dual optimization is then obtained by minimizing the opposite of the Lagrangian with respect to its dual variables:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\beta,\delta}{min} \quad & \boldsymbol{\alpha}^T\left(\frac{1}{4\lambda}\mathbf{G}' + \frac{1}{4\mu}\mathbb{1}_{n\times n}\right)\boldsymbol{\alpha} + \frac{\beta^2}{4\mu} + \frac{\delta^2}{4\mu} + \\ & \boldsymbol{\alpha}^T\left(diag(\mathbf{G}) - \left(B + \frac{\beta}{2\mu} - \frac{\delta}{2\mu}\right)\mathbb{1}_n\right) + \beta\left(B - \frac{\delta}{2\mu}\right), \\ s.t. \quad & 0 \leq \alpha_i \leq \frac{1}{n}, \quad \forall i = 1, \ldots, n, \\ & \beta, \delta \geq 0, \end{aligned} \quad (4)$$

where $\mathbf{G}$ is the Gram matrix defined by $G_{ij} = \langle (\boldsymbol{x}_i - \boldsymbol{c}), (\boldsymbol{x}_j - \boldsymbol{c}) \rangle$ and $\mathbf{G}'$ is the Hadamard product of $\mathbf{G}$ with itself. $\mathbb{1}_n$ (respectively $\mathbb{1}_{n \times n}$) represents a vector of length n (respectively a matrix of size $n \times n$) where entries are equal to 1.

3 Generalization Guarantees

One of our main contributions in this paper takes the form of a generalization guarantee on the algorithm ME^2. Since we learn a local model from a subset of training examples we need to prove the ability of ME^2 to perform well in generalization - that is - to exclude correctly new negative instances from the learned ellipsoid centered at this positive example. To do so, we derive in this section a generalization bound according to the theoretical framework of uniform stability (Bousquet and Elisseeff, 2002).

3.1 Uniform Stability

Roughly speaking, an algorithm is *stable* if its output does not change significantly under a small modification of the training sample. A formal definition is given below.

Definition 1. *(Bousquet and Elisseeff, 2002) A learning algorithm has a uniform stability in $\frac{\beta}{n}$ with respect to a loss function ℓ and a parameter set θ, with β a positive constant if:*

$$\forall S,\ \forall i,\ 1 \leq i \leq n,\ \sup_{\boldsymbol{x}} |\ell(\theta_S, \boldsymbol{x}) - \ell(\theta_{S^i}, \boldsymbol{x})| \leq \frac{\beta}{n},$$

where S is a learning sample of size n, θ_S the model parameters learned from S, θ_{S^i} the model parameters learned from the sample S^i obtained by replacing the i^{th} example $\boldsymbol{x}_i$ from S by another example $\boldsymbol{x}'_i$ independent from S and drawn from P. $\ell(\theta_S, \boldsymbol{x})$ is the loss suffered at $\boldsymbol{x}$.

One can then obtain the following generalization bound[1]:

Theorem 1 (from Bousquet and Elisseeff (2002), Thm 12)**.** *Let $\delta > 0$ and $n > 1$. For any algorithm with uniform stability β/n, using a loss function bounded by b, with probability $1 - \delta$ over the random draw of S:*

$$L(\theta_S) \leq \hat{L}_S(\theta_S) + \frac{2\beta}{n} + (4\beta + b)\sqrt{\frac{\ln 1/\delta}{2n}},$$

where $L(\cdot)$ is the true risk and $\hat{L}_S(\cdot)$ its empirical estimate over S.

3.2 Generalization Bound

Given a centroid c (representing a positive instance) and a learning sample $S = \{\boldsymbol{x}_i\}_{i=1}^n$ of negative instances drawn *i.i.d.* from an unknown probability distribution P_-, the set of parameters to be learned by ME^2 is the pair $(R, \mathbf{M})$. For convenience, we consider the following optimization problem that is equivalent to Problem 2:

$$\begin{array}{ll} \underset{R,\mathbf{M}}{min} & \frac{1}{n}\sum_{i=1}^n \ell(R, \mathbf{M}, \boldsymbol{x}_i) + \mu(B - R)^2 + \lambda\|\mathbf{M} - \mathbf{I}\|_F^2, \\ s.t. & B \geq R \geq 0. \end{array} \tag{5}$$

[1] If this result was proposed in the context of regression and classification tasks, the proof techniques are general enough - by considering generic bounded and lipschitz losses - so that it also holds for the setting considered in this section.

where $\ell(\cdot)$ represents the loss such that $\ell(R, \mathbf{M}, \boldsymbol{x}_i) = \frac{1}{n}[R - \|\boldsymbol{x}_i - \boldsymbol{c}\|_M^2]_+$ with $[\cdot]_+$ the hinge loss function: $[a]_+ = \max(a, 0)$.
The true risk is defined by $L(\mathbf{M}, R) = \mathbb{E}_{\boldsymbol{x}\sim P_-}\ell(\mathbf{M}, R, \boldsymbol{x})$ and its empirical estimate over the sample S by $\hat{L}_S(\mathbf{M}, R) = \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{M}, R, \boldsymbol{x}_i)$. We also denote the regularization term as $N(\mathbf{M}, R) = \mu(B - R)^2 + \lambda\|\mathbf{M} - \mathbf{I}\|_F^2$ and assume that data are bounded by K. F_S denotes the function to be minimized, *i.e.*:

$$F_S(\mathbf{M}, R) = \hat{L}(\mathbf{M}, R) + N(\mathbf{M}, R).$$

Note here that it can easily be checked that our loss function ℓ is convex with respect to $\mathbf{M}$ and R. To prove a generalization bound on our algorithm ME^2, we need to prove that our setting verifies the definition of uniform stability. For this purpose, we first prove that our loss function is actually k-lipschitz in its first two arguments.

Lemma 1. *The loss ℓ is k-lipschitz with respect to $\mathbf{M}$ and R with $k = \max(1, 4K^2)$, i.e.: for any $(\mathbf{M}, R)$, $(\mathbf{M}', R')$, $\forall \boldsymbol{x}$:*

$$|\ell(\mathbf{M}, R, \boldsymbol{x}) - \ell(\mathbf{M}', R', \boldsymbol{x})| \leq k\|(\mathbf{M}, R) - (\mathbf{M}', R')\|,$$

where $\|(\mathbf{M}, R) - (\mathbf{M}', R')\| = |R - R'| + \|\mathbf{M} - \mathbf{M}'\|_F$.

Proof. The proof uses successively the fact that the hinge loss is 1-lipschitz and a property of the absolute value. We then use the Cauchy-Schwarz inequality and classic properties on norms. □ □

We now need a technical lemma on the objective function F_S.

Lemma 2. *Let S be a learning sample, let F_S and F_{S^i} be two objective functions with respect to samples S and S^i and let $(\mathbf{M}, R)$ and $(\mathbf{M}^i, R^i)$ be their respective minimizers. We also define $\Delta(\mathbf{M}, R) = (\mathbf{M}^i, R^i) - (\mathbf{M}, R)$ and recall that $N(\mathbf{M}, R) = \mu(B - R)^2 + \lambda\|\mathbf{M} - \mathbf{I}\|_F^2$. We have, for all $t \in [0, 1]$:*

$$N(\mathbf{M}, R) - N((\mathbf{M}, R) + t\Delta(\mathbf{M}, R)) + N((\mathbf{M}^i, R^i) - N((\mathbf{M}^i, R^i) - t\Delta(\mathbf{M}, R)) \leq \frac{2t\max(1, 4K^2)}{n}\|\Delta(\mathbf{M}, R)\|.$$

Proof. The left hand side of the previous inequality can be written as follows:

$$\begin{aligned}\mu[(B - R)^2 + (B - R^i)^2 - (B - (R + t\Delta R))^2 - (B - (R^i - t\Delta R))^2] \\ + \lambda[\|\mathbf{M} - \mathbf{I}\|_F^2 + \|\mathbf{M}^i - \mathbf{I}\|_F^2 - \|\mathbf{M} + t\Delta\mathbf{M} - \mathbf{I}\|_F^2 - \|\mathbf{M}^i - t\Delta\mathbf{M} - \mathbf{I}\|_F^2] \\ = \mu\theta(R) + \lambda\tau(\mathbf{M}) \text{ (for the sake of simplification)} \quad (6)\end{aligned}$$

The upper bound is then given by using the same development as the one given in the proof of *lemma 20* from Bousquet and Elisseeff (2002). □ □

We are now able to prove the stability of our algorithm.

Proposition 1. *It exists a positive constant κ such that the algorithm ME^2 is uniformly stable with $\beta = \frac{2(\max(1, 4K^2))^2}{\kappa\min(\mu, \lambda)}$.*

Proof. We give here the main steps of the proof, using notations introduced before.
Setting $t = \frac{1}{2}$ we have from Lemma 2

$$\mu\theta(R) + \lambda\tau(\mathbf{M}) \leq \frac{\max(1, 4K^2)}{n}\|\Delta(\mathbf{M}, R)\|. \tag{7}$$

Then, by developping the left hand side of equation (7) we have:

$$\mu(R^i - R)^2 + \lambda\|\mathbf{M}^i - \mathbf{M}\|_F^2 \leq \frac{2\max(1, 4K^2)}{n}\|\Delta(\mathbf{M}, R)\|. \tag{8}$$

Recall that $\|\Delta(\mathbf{M}, R)\| = |R - R^i| + \|\mathbf{M}^i - \mathbf{M}\|_F$. Because we are working in a finite space, all the norms are equivalent, *i.e.* it exists a positive constant κ such that, $\forall(R, R^i) \in \mathbb{R}^+$, $\forall(\mathbf{M}, \mathbf{M}^i) \in \mathbb{R}^{d\times d}$ we have:

$$\kappa(|R - R^i| + \|\mathbf{M}^i - \mathbf{M}\|_F)^2 \leq (R - R^i)^2 + \|\mathbf{M}^i - \mathbf{M}\|_F^2. \tag{9}$$

The left hand side is equal to $\kappa\|\Delta(\mathbf{M}, R)\|^2$. Thus, by multiplying (9) by $min(\mu, \lambda)$ and using the inequality (8), we have:

$$\|\Delta(\mathbf{M}, R)\| \leq \frac{2\max(1, 4K^2)}{n\kappa\min(\mu, \lambda)}.$$

Finally, starting from the left-hand side of Definition 1 and applying Lemma 1 and the previous inequality leads to our result. □

□

It remains to show that our hinge-loss function ℓ is bounded, which is the case because the radius is bounded by B, so is ℓ.
Given the stability constant and the fact that the loss is bounded, using Theorem 1, we obtain our final result:

Theorem 2. *Let $\delta > 0$ and $n > 1$. There exists a constant $\kappa > 0$, such that with probability at least $1 - \delta$ over the random draw over S, we have for any $(\mathbf{M}, R)$ solution of Problem 5:*

$$L(\mathbf{M}, R) \leq \hat{L}_S(\mathbf{M}, R) + \frac{4(\max(1, 4K^2))^2}{n\kappa\min(\mu, \lambda)} + \left(\frac{8(\max(1, 4K^2))^2}{\kappa\min(\mu, \lambda)} + B\right)\sqrt{\frac{\ln 1/\delta}{2n}}.$$

This generalization bound holds for any positive center $\boldsymbol{c}$. If one has p positive centers, by the union bound, we can extend the previous result for each of the p centers with probability $1 - \delta/p$ showing that the learned ellipsoids can exclude negative instances with high probability. We can notice that the bound implicitly depends on the dimension of the data through the hyperparameters μ and λ.

4 Experiments

4.1 Algorithms and Datasets

In this section, we aim at evaluating the behavior of ME^2 in comparison to some state of the art algorithms. Those methods have been selected to characterize some specificities of ME^2.
• Since we established a link between our local ellipsoids and the rules induced by decision trees in the form of local rectangles (Figure 2), we compare ME^2 with standard **decision trees** (DT). The objective here is to show that the *learned ellipsoids better capture the local information* of the input

Table 1: Number of instances, number of features, Imbalance Ratio (i.e. number of positives over the number of instances).

Dataset	Nb. of ex.	Nb. of feat.	IR
Yeast3	1 484	8	10.9%
Abalone	4 177	8	10.7%
Wine	1 599	11	3.3%
Abalone 17	2 338	8	2.5%
Yeast6	1 484	8	2.4%
Abalone 20	1 916	8	1.4%
Bank Fraud	15 000	17	1%

space.

• To deal with imbalanced datasets, a commonly used strategy consists of sampling the data to fix the imbalancy problem. Therefore, we also learn a decision tree DT_O (resp. DT_U) after a pre-processing step which consists of **oversampling** (with replacement) the minority class examples (resp. **undersampling** the majority class examples). We also combine the two previous approaches (DT_{OU}). Finally, we apply a SMOTE-like strategy (Chawla et al., 2002) (DT_{SMOTE}) which creates synthetic minority class examples in the neighborhood of the positive data. The goal of this comparison is to show how ME^2 behaves even if *it does not resort to sampling processes.*

• When the proportion of positive examples is too small, some Support Vector Data Description methods (SVDD) (Azami et al., 2014; Pauwels and Ambekar, 2011; Tax and Duin, 2004) - like **one-class SVMs** (Heller et al., 2003) - address the anomaly detection problem as an unsupervised outlier detection task. We run here one-class SVMs with two kernels: a linear kernel (LOCSVM) and a RBF kernel (KOCSVM). The objective of this comparison is to check if ME^2 *makes a good use of the few positive labels* compared to unsupervised methods. We also make use of the labels and run standard linear SVMs (LSVM), and RBF kernel-based SVMs (KSVM).

All the classifiers are trained using the corresponding machine learning packages in R^2, that is **C50** for the decision trees, **e1071** for the SVMs, **DMwR** for SMOTE and **Rsolnp** for ME^2.

The experiments are performed on 6 datasets coming from the UCI and KEEL databases[3] and one proprietary database on a bank fraud detection task. Their characteristics (number of examples, features, imbalance ratio IR) are described in Table 1. Note that the categorical variables have been replaced by binary features.

4.2 Experimental setup

As explained in the introduction, the classic *accuracy* is not a suitable criterion to address issues due to the presence of imbalanced data. For this reason, we evaluate the algorithms using the *F-measure* defined as the harmonic mean of the *Precision* and *Recall* criteria, where *F-measure* $= \frac{2\times Precision\times Recall}{Precision+Recall}$ with $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, where TP is the number of true positives, FP the number of false positives and TN the number of true negatives.

For each series of experiments, the dataset is divided into 80%/20%. A 2-fold cross-validation is applied on the first set S (while preserving the same IR in each fold) to tune the hyperparameters.

[2]https://www.r-project.org/

[3]These datasets can be found either on the *UCI Repository* (https://archive.ics.uci.edu/ml/datasets.html) or the *KEEL* website (http://sci2s.ugr.es/keel/imbalanced.php?order=ir#sub60).

Table 2: Comparison of the methods in terms of F-Measure over 10 runs. The best results are indicated in bold font. The last column reports the average running time (in sec) for one run.

Algorithm	Yeast3	Abalone	Wine	Abalone17	Yeast6	Abalone20	Bank Fraud	Time
DT	**0.77 ± 0.06**	0.64 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	**0.51 ± 0.23**	0.10 ± 0.15	0.00 ± 0.00	3.2
DT_O	0.75 ± 0.06	0.64 ± 0.04	0.06 ± 0.09	0.35 ± 0.08	0.45 ± 0.12	**0.30 ± 0.15**	0.04 ± 0.03	3.5
DT_U	0.76 ± 0.08	**0.67 ± 0.03**	0.09 ± 0.11	0.28 ± 0.11	0.49 ± 0.11	0.19 ± 0.20	0.04 ± 0.03	2.4
DT_{OU}	0.72 ± 0.04	0.61 ± 0.04	0.15 ± 0.06	0.34 ± 0.05	0.36 ± 0.12	**0.30 ± 0.12**	**0.05 ± 0.04**	2.6
DT_{SMOTE}	0.65 ± 0.10	0.57 ± 0.07	0.14 ± 0.12	0.24 ± 0.09	0.18 ± 0.09	0.27 ± 0.13	0.04 ± 0.03	28.1
LSVM	0.67 ± 0.05	0.62 ± 0.01	0.14 ± 0.08	0.29± 0.01	0.30 ± 0.05	0.23 ± 0.02	0.03 ± 0.02	702.5
RBFSVM	0.66 ± 0.09	0.63 ± 0.03	0.07 ± 0.08	0.17 ± 0.06	0.36 ± 0.09	0.13 ± 0.13	0.00 ± 0.00	39.2
LOCSVM	0.01 ± 0.02	0.11 ± 0.03	0.02 ± 0.07	0.10 ± 0.05	0.00 ± 0.00	0.05 ± 0.13	0.03 ± 0.01	28.3
KOCSVM	0.01 ± 0.02	0.21 ± 0.04	0.01 ± 0.07	0.06 ± 0.03	0.00 ± 0.00	0.05 ± 0.11	0.00 ± 0.00	1472.0
ME^2	0.67 ± 0.06	0.61 ± 0.03	**0.20 ± 0.09**	**0.46 ± 0.07**	0.46 ± 0.08	**0.30 ± 0.07**	**0.05 ± 0.02**	2.9

The second set is used as a test set. Each experiment is repeated 10 times and the reported results are the averages over the 10 trials.

Remember that we learn an ellipsoid centered at each positive example of S. This ellipsoid defines in some way the local region of the projection space which is under the influence of the considered positive example. In this context, at both validation (to tune the parameters) and test time, a query $\boldsymbol{x}'$ is associated to its closest positive example $N_{\boldsymbol{x}'}$ (with respect to the Euclidean distance) in the training set. Then, in order to take into account the local density of positives and negatives in the corresponding ellipsoid, and following the idea suggested in Barandela et al. (2003), we apply the following decision rule. $\boldsymbol{x}'$ will be predicted as positive if: 1) It is inside the ellipsoid centered at $N_{\boldsymbol{x}'}$. This means that $\boldsymbol{x}'$ is actually under the influence of $N_{\boldsymbol{x}'}$ which occurs when the corresponding learned Mahalanobis distance verifies: $\|\boldsymbol{x}' - N_{\boldsymbol{x}'}\|_{\mathbf{M}_{N_{\boldsymbol{x}'}}} \leq R_{N_{\boldsymbol{x}'}}$, where $\mathbf{M}_{N_{\boldsymbol{x}'}}$ is the PSD matrix learned by ME^2 corresponding to the ellipsoid centered at $N_{\boldsymbol{x}'}$ and $R_{N_{\boldsymbol{x}'}}$ is its associated radius. 2) Its nearest neighbor in the ellipsoid is a positive example with respect to the learned local distance $\|\boldsymbol{x}' - \boldsymbol{x}\|_{\mathbf{M}_{N_{\boldsymbol{x}'}}}$. Otherwise, $\boldsymbol{x}'$ is predicted as negative.

Note that the hyper-parameters μ and λ are tuned respectively in the range $\{0.75, 0.8, 0.85, 0.9, 0.95, 1, 2, 10\}$ and $\{10^{-6:2}\}$ by maximizing the F-Measure for each local model according to the previous rule.

4.3 Results

The results are reported in Table 2. The datasets are sorted from the least to the most imbalance ratio to see the effect of ME^2 with a decreasing rate of positive examples. We can make the following remarks:

• On average, ME^2 outperforms all the other methods even if it does not resort to sampling processes. As shown in Figure 3, its average rank over the 7 datasets (**2.6**) is better than the others. If we focus on the 5 datasets with large imbalancy (*Wine*, *Abalone17*, *Yeast6*, *Abalone20* and the *Bank Fraud*), ME^2 is even better with an average rank of **1.4**.

• For the first two datasets (i.e. *Yeast3* and *Abalone*) where the rate of positive examples is greater than 10%, our method is not very useful. This behavior can be explained by two reasons: (i) when the number of learned ellipsoids grows, their overlapping is larger and larger and therefore the False Positive rate increases; (ii) a large number of ellipsoids induces an increasing risk of generating close ellipsoids with different orientations and shapes. About this second point, an interesting perspective would consist of constraining close positive examples to have similar ellipsoids.

• Compared with decision trees, these experiments show that ME^2 has a much better capacity to capture the local specificities of the feature space than the local rectangles learned by decision trees.

For three datasets, it is worth noticing that decision trees even do not capture anything (*Wine*, *Abalone17*, *Abalone20*).
• The results obtained by one-class SVMs are much worse than the other methods. This behavior shows that for all the datasets, including the bank fraud database, the positive examples cannot be considered as outliers and that their underlying distribution is likely to be multimodal.
• ME^2 works better than SVMs while the latter use a reweighting scheme in the objective function to balance the data.

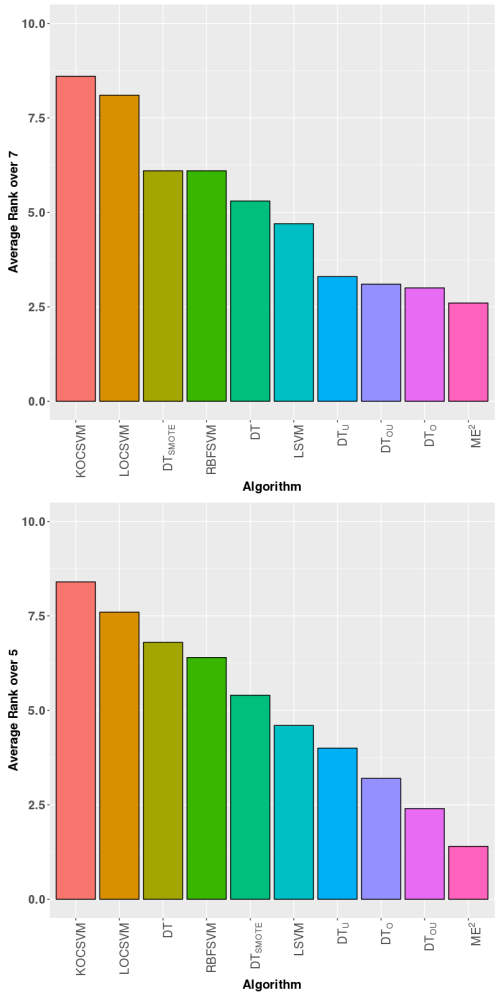

Figure 3: Average ranks over all the datasets (top) and over the five most imbalanced datasets (bottom).

We also report in Table 2 the average running time for one run of each method. Note that since ME^2 can be parallelized, we report the running time required for tuning and solving the optimization problem 2. We can see that since ME^2 learns matrices that directly satisfy the positive semi definiteness, our method is efficient, *i.e.* very close to decision trees. However, note that if one uses ME^2 without parallelizing the learning of the p ellipsoids, the running time will be on average p times the result reported in Table 2. But ME^2 will be still efficient (at least better then kernelized-SVMs) since p is supposed to be very small in highly imbalanced scenarios.

5 Conclusion

We have presented a method to learn *Maximum Excluding Ellipsoids* in the context of imbalanced binary classification tasks. Our algorithm, called ME^2, is based on simple local linear models. Moreover, we have proven its uniform stability which takes the form of a generalization bound on the learned matrix $\mathbf{M}$. We have shown that our method is particularly efficient and robust when the rate of positive examples is very small. The reason comes from the fact that ME^2 is able to learn decision boundaries in the form of ellipsoids (via a metric learning-based strategy) that are optimized locally to better fit the specificities of the space. ME^2 is based on a very simple decision rule looking for the nearest ellipsoid to a test query. We think that this rule may benefit from further investigation, e.g. by considering a combination of ellipsoids to predict the label of a test data. This would be possible by using a graph over the ellipsoids centers where information would be shared like in an information network. Besides, from a theoretical point of view, we have derived a guarantee on the learned matrix $\mathbf{M}$ and radius R. Since our decision rule is close to a nearest neighbor classifier decision rule, it would be interesting to establish a link between the quality of $\mathbf{M}$ and R and the generalization error of such a classifier. Another perspective would be to partition the positive example space and constrain the ellipsoids to be similar in terms of orientation (using some regularization) if they have been learned from the same cluster. Finally, in a context of fraud detection where the fraud strategy tends to evolve through time, developing an online version of our algorithm might be relevant to better capture distribution shifts.

References

Aggarwal, C. C. (2013). *Outlier Analysis*. Springer.

Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. In *19th International Conference in Knowledge Based and Intelligent Information and Engineering Systems KES*, volume 60 of *Procedia Computer Science*, pages 708 – 713. Elsevier.

Azami, M. E., Lartizien, C., and Canu, S. (2014). Robust outlier detection with L0-SVDD. In *22th Eu Symposium on Artificial Neural Net. (ESANN)*.

Barandela, R., Sánchez, J. S., García, V., and Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851.

Bellet, A., Habrard, A., and Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709.

Bellet, A., Habrard, A., and Sebban, M. (2015). *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Boujnouni, M. E., Jedra, M., and Zahid, N. (2012). New decision function for support vector data description. In *Snd Int. Conf. on Innovative Computing Technology (INTECH 2012)*, pages 305–310.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2:499–526.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Frery, J., Sebban, M., Habrard, A., Caelen, O., and Guelton, L. (2017). Efficient top rank optimization with gradient boosting for supervised anomaly detection. In *ECML-PKDD 2017*.

Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11 4:e0152173.

Heller, K. A., Svore, K. M., Keromytis, A. D., and Stolfo, S. J. (2003). One class support vector machines for detecting anomalous windows registry accesses. In *ICDM work. on Data Min. for Computer Security*.

Khalilia, M., Chakraborty, S., and Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, 11(1):51.

Le, T., Tran, D., and Ma, W. (2013). Fuzzy multi-sphere support vector data description. In *17th Pacific-Asia Conference (PAKDD), Part II*, pages 570–581. Springer.

Liu, Y. and Zheng, Y. (2006). Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *18th IEEE International Conference on Pattern Recognition (ICPR06)*.

Pauwels, E. J. and Ambekar, O. (2011). One class classification for anomaly detection: Support vector data description revisited. In *Industrial Conference on Data Mining*, pages 25–39.

Perrot, M. and Habrard, A. (2015). Regressive virtual metric learning. In *NIPS*.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Shi, Y., Bellet, A., and Sha, F. (2014). Sparse compositional metric learning. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 2078–2084.

Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Machine Learning Journal*, 54(1):45–66.

Wang, Z., Gao, D., and Pan, Z. (2010). An effective support vector data description with relevant metric learning. In *7th International Symposium on Neural Networks (ISNN), Part II*, pages 42–51. Springer.